Mitglied der Helmholtz-Gemeinschaft

JÜLICH FORSCHUNGSZENTRUM

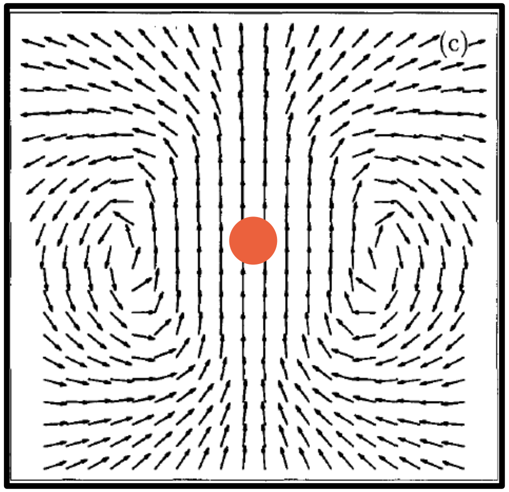
Multi-Particle Collision Dynamics on Massively Parallel Computers

Godehard Sutmann
Institute for Advanced Simulation (IAS) - Jülich Supercomputing Centre (JSC)
Research Centre Jülich

g.sutmann@fz-juelich.de

Hydrodynamic Interactions

- Force field description includes direct interactions between particles
- Interactions may also be mediated by a medium between particles
- Moving particle in a fluid will create a velocity field which interact with particles far apart
- Proper description for velocity fields in a fluid are the Navier-Stokes equations



$$\rho\partial_t \mathbf{v} + \rho(\mathbf{v}\nabla)\,\mathbf{v} + \nabla p - \eta_{sh}\nabla^2\mathbf{v} - (\eta_{sh} + \eta_{dil})\nabla(\nabla\mathbf{v}) = \mathbf{f}$$

$$\partial_t\rho + \rho\nabla\mathbf{v} = 0$$

Hydrodynamic Interactions

- For small Reynolds numbers (velocities) the NS equations may be linearized

$$\rho\partial_t \mathbf{v} + \nabla p - \eta_{sh}\nabla^2\mathbf{v} = \mathbf{f}$$

- The velocity field is then given by a “response” function to the external perturbation

$$\mathbf{v}(\mathbf{r}) = \mathsf{T}(\mathbf{r},\mathbf{r}')\mathbf{f}(\mathbf{r}')$$

- The Green's function is the Oseen tensor (not positive definite)

$$\mathsf{T}^{(O)}(r) = \frac{1}{8\pi\eta_{sh}r}\left(\mathbf{1} + \hat{\mathbf{r}}\hat{\mathbf{r}}\right)$$

long ranged

- Extension is the Rotne-Prager tensor (positive definite)

$$\mathsf{T}^{(RP)}(r) = \frac{1}{6\pi\eta_{sh}r}\left(\frac{3}{4}(\mathbf{1} + \hat{\mathbf{r}}\hat{\mathbf{r}}) + \frac{1}{2}\frac{a^2}{r^2}(\mathbf{1} - 3\hat{\mathbf{r}}\hat{\mathbf{r}})\right)$$

Hydrodynamic Interactions

- This description is a simplified view and only valid for dilute solutions
- No thermodynamic fluctuations included
- Hydrodynamic interactions are not pairwise additive
- Leads to involved calculations if one wants to do it correctly
- More rigorous calculations would solve Navier-Stokes equations with solvated particles
  - problem of dynamic boundary conditions at particle positions
  - problem of discretisation of compute domain at particle positions
  - limitation to only a few particles
- Approximate treatment within Stokesian Dynamics

Discrete schemes for hydrodynamics

- To facilitate treatment of hydrodynamic interactions, several schemes were developed
  - Lattice-Boltzmann
  - Dissipative Particle Dynamics
  - Smooth Particle Hydrodynamics
  - Stochastic Rotation Dynamics / Multi-Particle Collision Dynamics

Multi-Particle Collision Dynamics

- Solvent particles are considered to exchange momentum between solvent and solute particles
  - Momentum exchange occurs after collisions
  - No explicit interactions (no hydrodynamic force field)
  - Microscopic details of solvent collisions are not of interest
  - Therefore modeled by stochastic momentum exchange

- Requirements
  - Conservation of energy
  - conservation of momentum
  - (conservation of angular momentum)

Multi-Particle Collision Dynamics

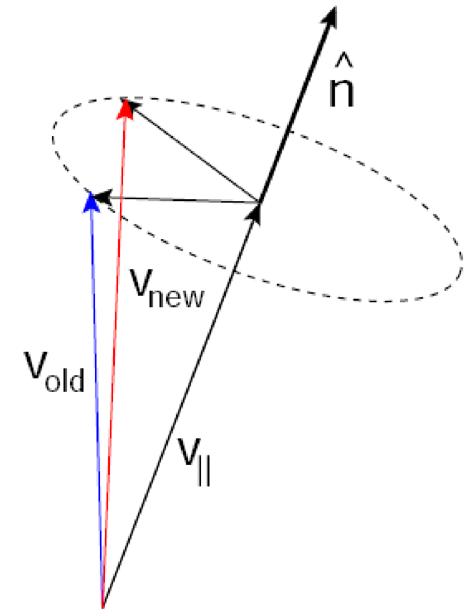
- Consider local environment of particles and allow for effective collisions
- sort particles into collision cells (density $\rho \in [5,20]$/cell)
- express particle velocities through projected parts onto a random axis (in each collision cell)

$$\mathbf{v}_{c,cm} = \frac{1}{M_c} \sum_{i \in C} m_i \mathbf{v}_i \qquad M_c = \sum_{i \in C} m_i \qquad \tilde{\mathbf{v}}_i = \mathbf{v}_i - \mathbf{v}_{c,cm}$$

with

$$\tilde{\mathbf{v}}_i = \tilde{\mathbf{v}}_{i,\parallel} + \tilde{\mathbf{v}}_{i,\perp} \qquad \tilde{\mathbf{v}}_{i,\parallel} = \hat{\mathbf{n}} \left(\hat{\mathbf{n}} \tilde{\mathbf{v}}_i \right) \qquad \tilde{\mathbf{v}}_{i,\perp} = \tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_{i,\parallel}$$

MPC collision: random rotation



- Perform a random rotation of relative velocities by

$$\tilde{\mathbf{v}}_i \mapsto \tilde{\mathbf{v}}_{i,||} + R(\tilde{\mathbf{v}}_{i,\perp}; \alpha)$$

with $\quad R(\mathbf{v};\alpha) = \mathbf{v}\ \cos 2\pi\alpha + (\mathbf{v} \times \hat{\mathbf{n}})\ \sin 2\pi\alpha$

- for every collision cell calculate a random rotation axis

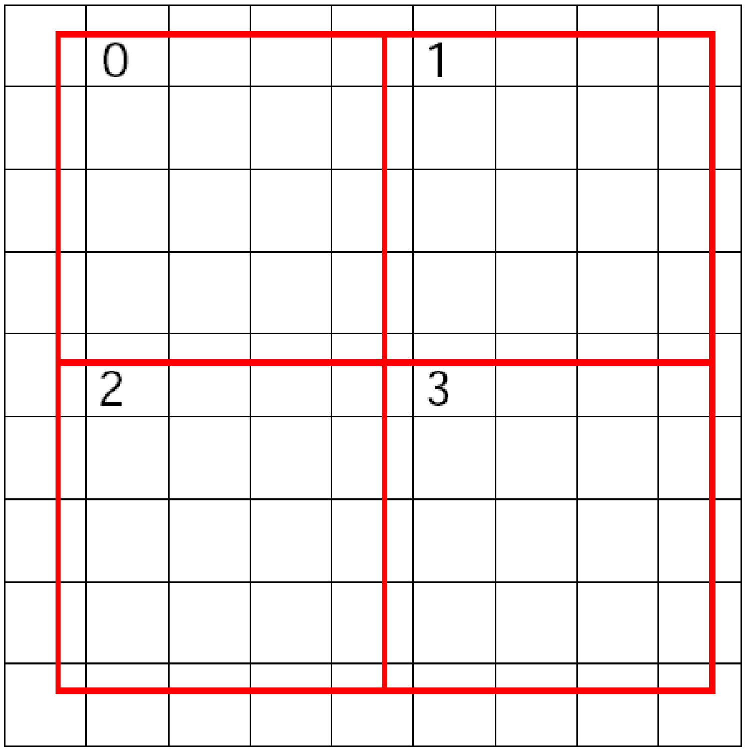
$$\hat{n}_x = \sqrt{1-(2\xi_1-1)^2}\cos 2\pi\xi_2 \qquad \hat{n}_y = \sqrt{1-(2\xi_1-1)^2}\sin 2\pi\xi_2$$

$$\hat{n}_z = 2\xi_1 - 1$$

- add com velocity of the cell to rotated relative velocities to get new particle velocities *after* a collision

$$\mathbf{v}_i \mapsto \tilde{\mathbf{v}}_i + \mathbf{v}_{c,cm}$$

MPC construction of collision cells



- particles are sorted into small cells in the system
- to conserve Galilean invariance, cells are shifted by a random offset in every time step

$$\delta L_{c,\alpha} = \xi_\alpha L_{c,\alpha} \qquad \xi_\alpha \mapsto \mathrm{U}[0,1]$$

- cell indices of the particles are calculated via

$$i_{c,\alpha} = \mathrm{int}\left(\frac{r_\alpha - r_{d0,\alpha} + \delta L_{c,\alpha}}{L_{c,\alpha}}\right)$$

It is equivalent to shift ALL particles by a random offset, before the collision step and shift them back after the random rotation

For the parallel implementation this would induce large traffic across domain boundaries

MPC: conservation properties

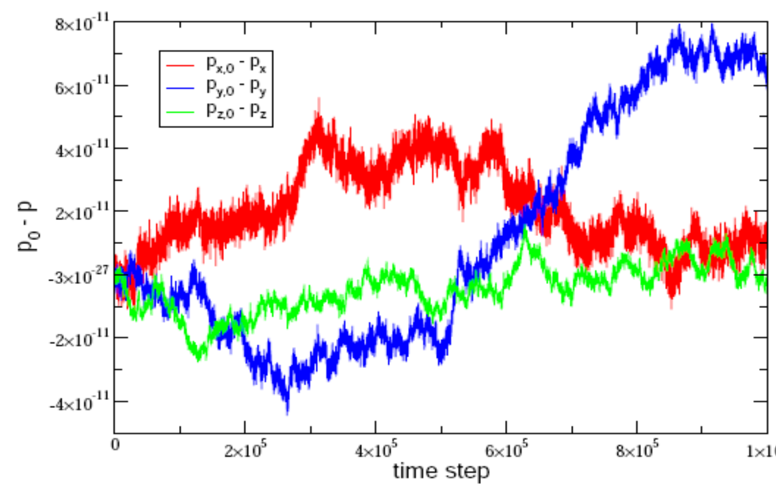
- Energy conservation (kinetic, since no potential)

$$
\begin{aligned}
K &= \frac{1}{2}\sum_{i=1}^{N} m_i \mathbf{v}_i^2 = \frac{1}{2}\sum_c \sum_{j\in c} m_j(\mathbf{v}_{c,cm}^2 + 2\mathbf{v}_{c,cm}\tilde{\mathbf{v}}_j + \tilde{\mathbf{v}}_j^2) \\
&= \frac{1}{2}\sum_c \sum_{j\in c} m_j(\mathbf{v}_{c,cm}^2 + \underbrace{2\mathbf{v}_{c,cm}\tilde{\mathbf{v}}_j}_{=0} + \tilde{\mathbf{v}}_j^T R^T R \tilde{\mathbf{v}}_j) \\
&= \frac{1}{2}\sum_c \sum_{j\in c} m_j(\mathbf{v}_{c,cm} + R\tilde{\mathbf{v}}_j)^2 = K
\end{aligned}
$$

- Momentum conservation

$$
\mathbf{P} = \sum_{i=1}^{N} m_i \mathbf{v}_i = \sum_c \sum_{j\in c} m_{j,c}\mathbf{v}_{j,c}
$$

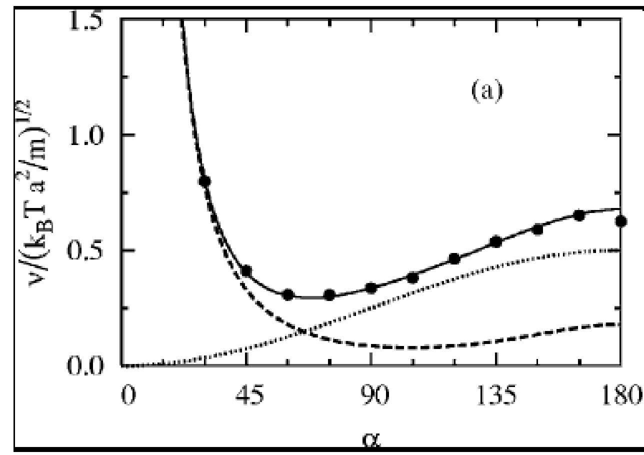
$$
= \sum_c M_c \mathbf{v}_{c,cm} + \sum_c \underbrace{\sum_{j\in c} m_{j,c}\tilde{\mathbf{v}}_{j,c}}_{=0}
$$



MPC: theoretical background

- it may be shown that MPC represents a discretised description of linearized Navier-Stokes equations

- theoretical expressions for transport coefficients
  - viscosity of the fluid $\eta = \eta_{kin} + \eta_{col}$



$$\eta_{coll} = (\gamma - 1 + e^{-\gamma}) \frac{m\,[1-\cos\alpha]}{12a\delta t}$$

$$\eta_{kin} = \rho k_B T \delta t \left(\frac{1}{1-w} - \frac{1}{2} \right)$$

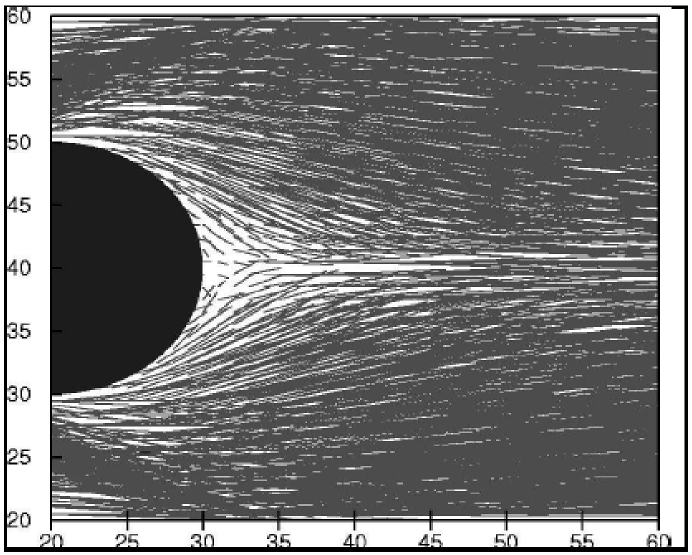
$$w = 1 - \left(\frac{\gamma - 1 + e^{-\gamma}}{5\gamma} \right) [4 - 2\cos\alpha - 2\cos 2\alpha]$$

γ	numb. part./cell
m	particle mass
α	rotation angle
a	coll. cell size
δt	time step

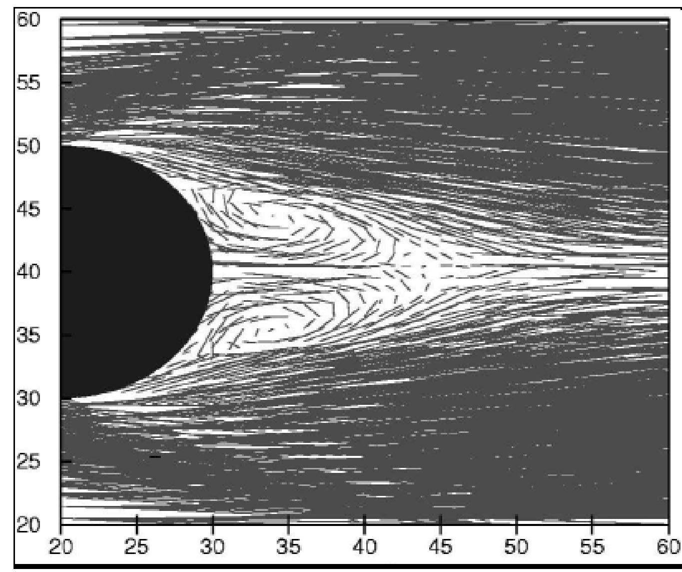
A.Malevanets and R.Kapral, J.Chem.Phys. **110**, 8605-8613 (1999)

MPC: numerical experiments

- Flow simulations around a cylinder



Re ~20



Re ~70

Alahyarov, Gompper, Phys.Rev.E **66**, 036702 (2002)

Solute/Solvent interactions: coupling to MD

- Need for combination between stochastic and deterministic methods
- Hydrodynamic modes evolve on a longer time scale than molecular modes

Approach:

- Perform usual molecular dynamics moves according to given interaction potentials
- Every n_{coll} timesteps include MD particles into a collision step

OR

- Do excluded volume simulations and calculate momentum transfer from solvent particles onto solute particles (like in hard sphere simulations)

Solute/Solvent interactions: coupling to MD

- Every n MD steps sort solutes together with solvent particles into collision cells

- Calculate common com-velocity

$$\mathbf{v}_{c,cm} = \frac{1}{M_C^{(A)} + M_C^{(B)}} \left(\sum_{a \in C}^{N_A} m_a \mathbf{v}_a + \sum_{b \in C}^{N_B} m_b \mathbf{v}_b \right)$$

$$\tilde{\mathbf{v}}_i = \mathbf{v}_i - \mathbf{v}_{c,cm} \qquad i \in \{A, B\}$$

  and perform a rotation in velocity space for both solvent and solute particles around the same random axis

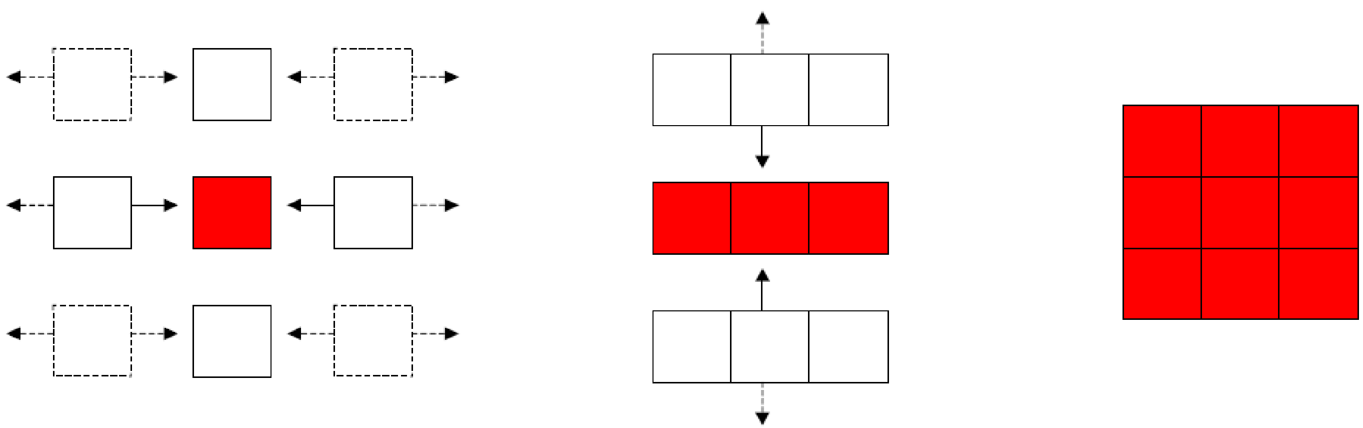
- Results in a stochastic momentum change of the solutes while conserving the overall momentum and kinetic energy in the system

- Inclusion of hydrodynamic modes into solute dynamics

Parallelization

Domain Decomposition

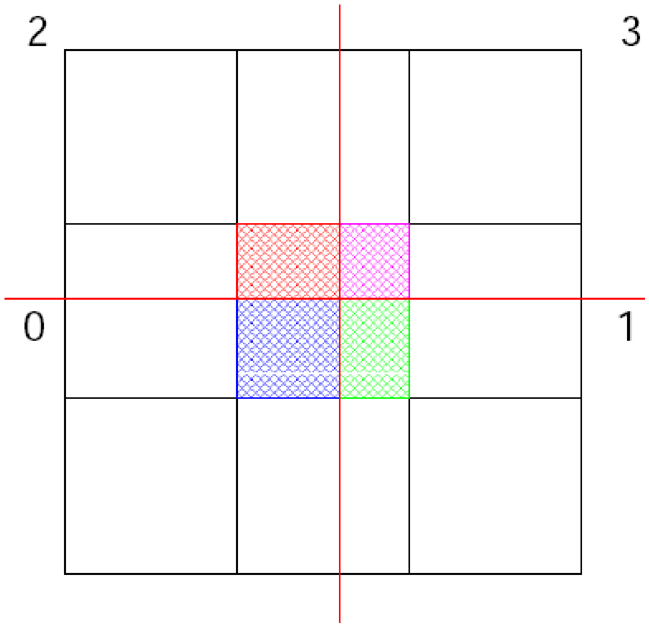
- Method is local – no long range interactions
- Every processor is responsible for a certain region
  Domain Decomposition
- In 3 dimensions every domain has 26 neighbors
- when particles leave domain: exchanged data are **x**, **v**, (m), i
- Efficient communication finishes after 2d steps:
  - send/receive left-right (x), out-in (y), bottom-top (z)



Domain Decomposition: cell exchange

- due to the random shift, collision cells will usually overlap with domain boundaries

- Requirements
  - com-velocities for overlapping cells
  - unique set of random numbers to set up the random rotation axis
- exchanged data: (partial) $\mathbf{v}_{cm}$, M_{cell}



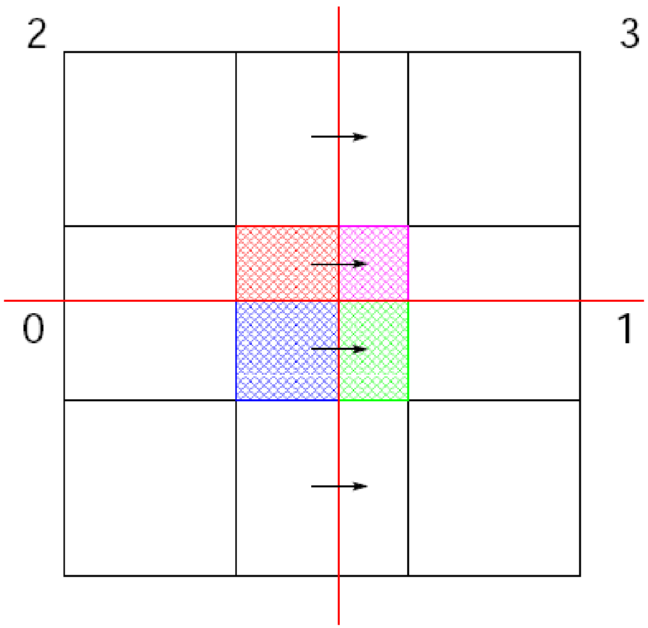
Collision cells may belong to

- 4 processors in 2-d
- 8 processors in 3-d

Domain Decomposition: cell exchange

- due to the random shift, collision cells will usually overlap with domain boundaries

- Requirements
  - com-velocities for overlapping cells
  - unique set of random numbers to set up the random rotation axis
- exchanged data: (partial) $\mathbf{v}_{cm}$, M_{cell}



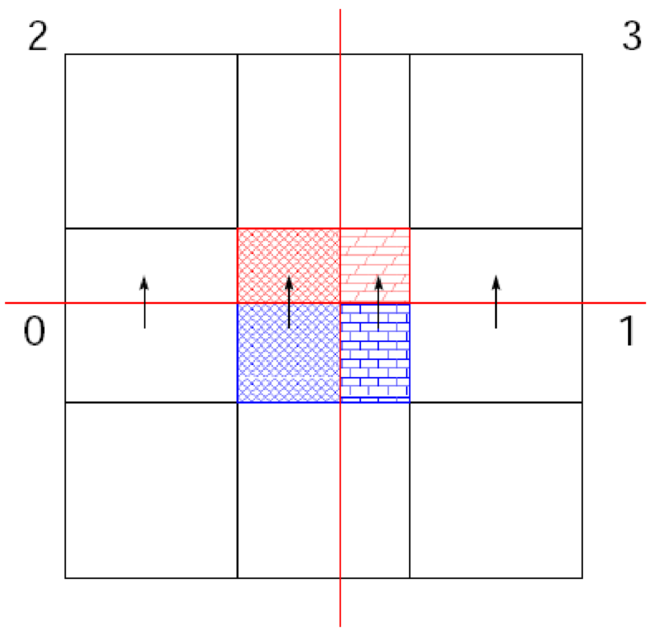
- 1st step:

send com-velocities of boarder cells to the right

Domain Decomposition: cell exchange

- due to the random shift, collision cells will usually overlap with domain boundaries

- Requirements
  - com-velocities for overlapping cells
  - unique set of random numbers to set up the random rotation axis
- exchanged data: (partial) $\mathbf{v}_{cm}$, M_{cell}



- 2nd step:

add received com-velocities

$$\mathbf{v}_{cm}^{(n')} = \frac{1}{M^{(n')}} \left(M_c^{(n)} \mathbf{v}_{c,cm}^{(n)} + M_c^{(m)} \mathbf{v}_{c,cm}^{(m)} \right)$$

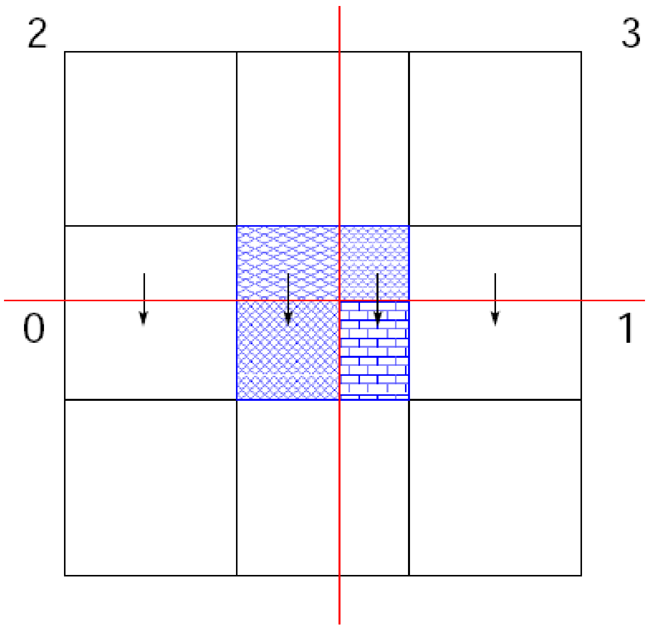
$$n' = n + m$$

and send them to the top

Domain Decomposition: cell exchange

- due to the random shift, collision cells will usually overlap with domain boundaries

- Requirements
  - com-velocities for overlapping cells
  - unique set of random numbers to set up the random rotation axis
- exchanged data: (partial) $\mathbf{v}_{cm}$, ξ_l



- 3rd step:

add received data results in total com-velocity of the cell

$$\mathbf{v}_{c,cm} = \frac{1}{M_c} \left(M_c^{(n')} \mathbf{v}_{cm}^{(n')} + M^{(l)} \mathbf{v}_{cm}^{(l)} \right)$$

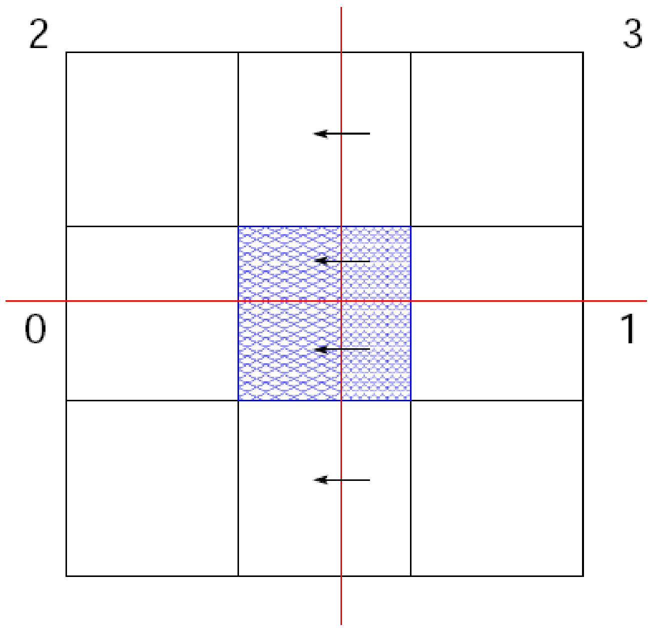
draw set of random numbers $\xi_\alpha \mapsto \mathrm{U}[0,1]$

send com-vel. and ξ to the bottom

Domain Decomposition: cell exchange

- due to the random shift, collision cells will usually overlap with domain boundaries

- Requirements
  - com-velocities for overlapping cells
  - unique set of random numbers to set up the random rotation axis
- exchanged data: (partial) $\mathbf{v}_{cm}$, ξ_l



- 4th step:

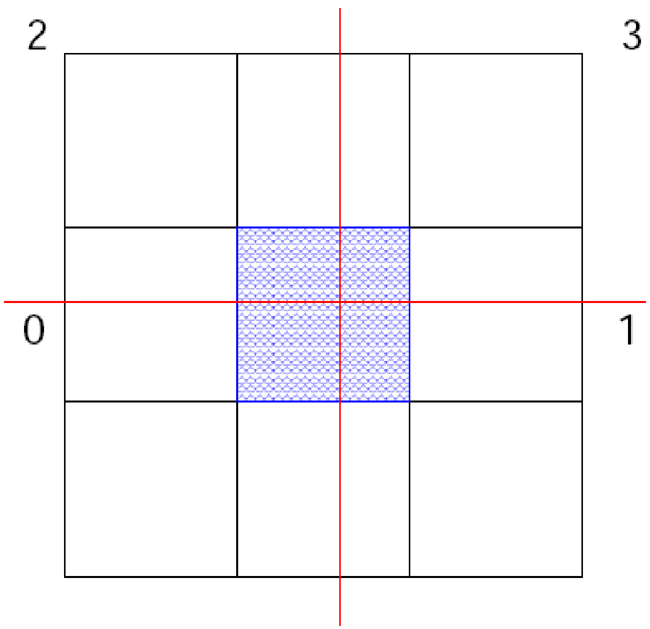
  receive random numbers and com-velocity of the cells and overwrite local information

  send com-velocity and random numbers to the left

Domain Decomposition: cell exchange

- due to the random shift, collision cells will usually overlap with domain boundaries

- Requirements
  - com-velocities for overlapping cells
  - unique set of random numbers to set up the random rotation axis
- exchanged data: (partial) $\mathbf{v}_{cm}$, ξ_l



- 5th step:

  receive random numbers and com-velocity of the cells and overwrite local information

  cell information is now unique across domain boundaries

Benchmarks

JÜLICH
FORSCHUNGSZENTRUM

Hardware

- MP2C is not designed for special hardware platform
- Running with MPI on several machines
  (IBM p690, CRAY XT4/5, IBM BG/P)
- Target platform for present benchmark:

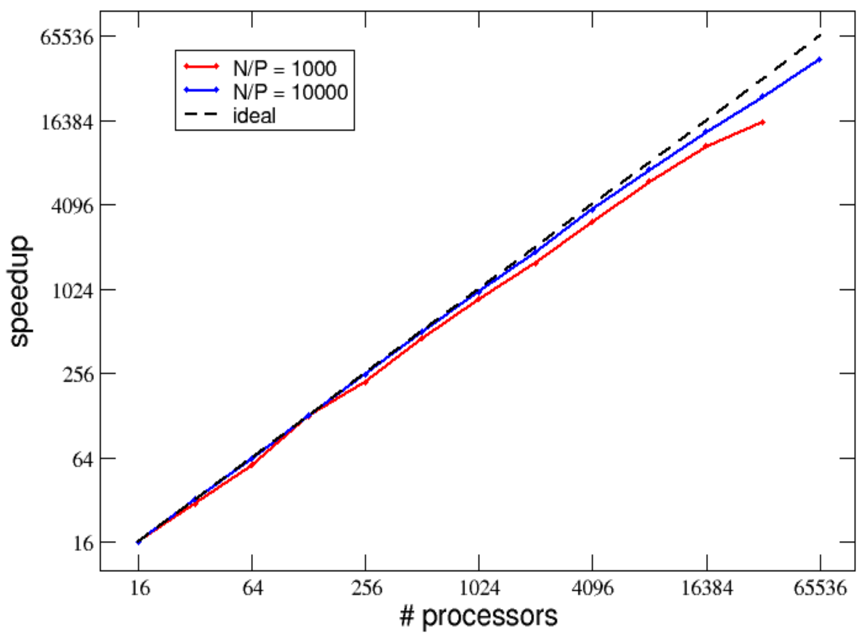
  IBM BlueGene/P at Jülich Supercomputing Centre

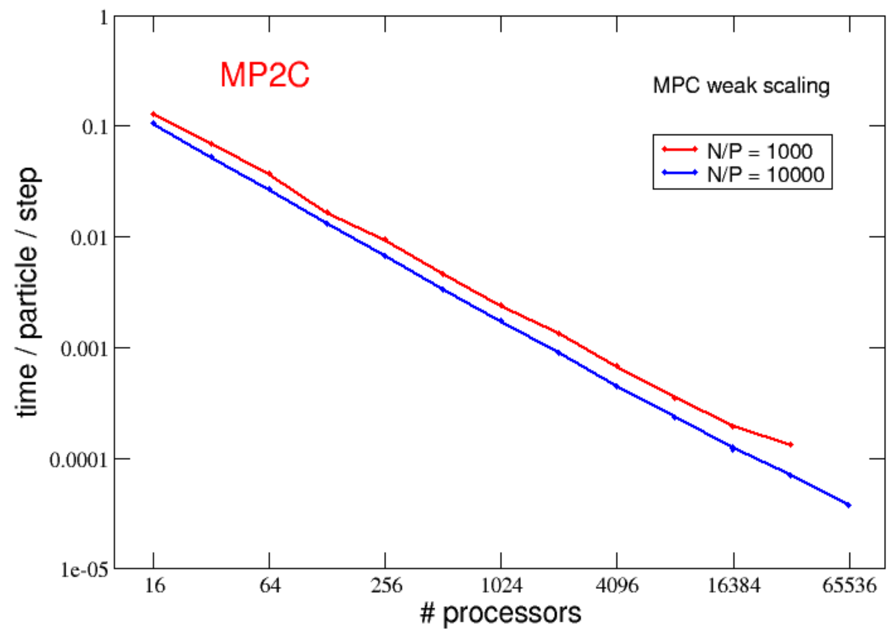
65,536 PEs, 4 PEs/node

223 Tflop/s peak,
180 Tflop/s Linpack
32 TByte memory
2 GByte per node

MPC: Weak scaling

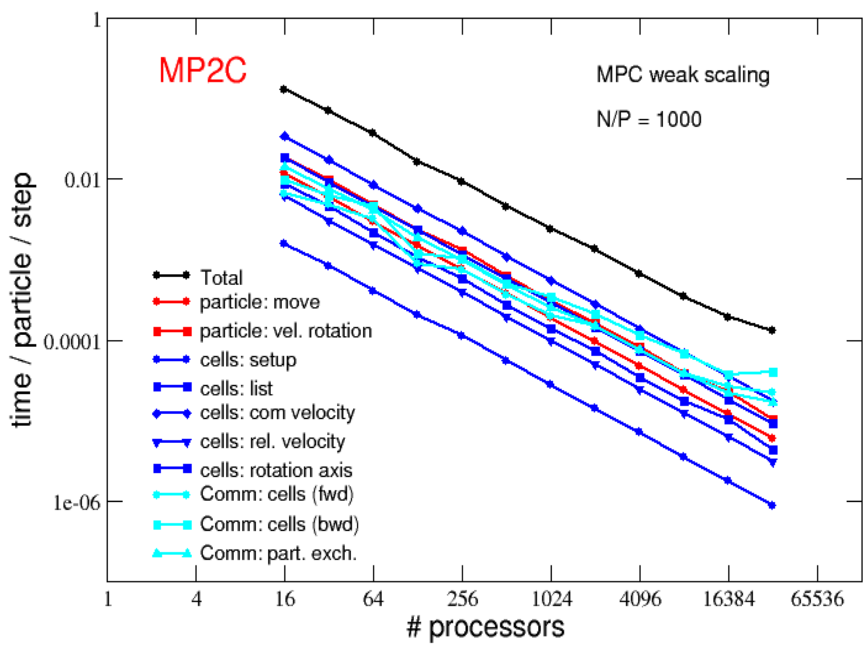
- Keep number of particles/processor constant

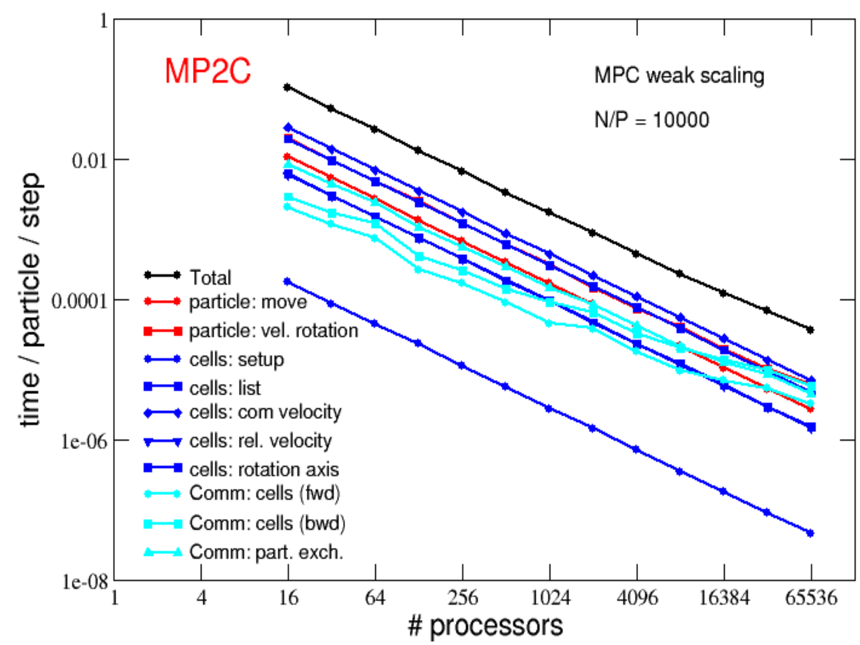




MPC: Weak scaling

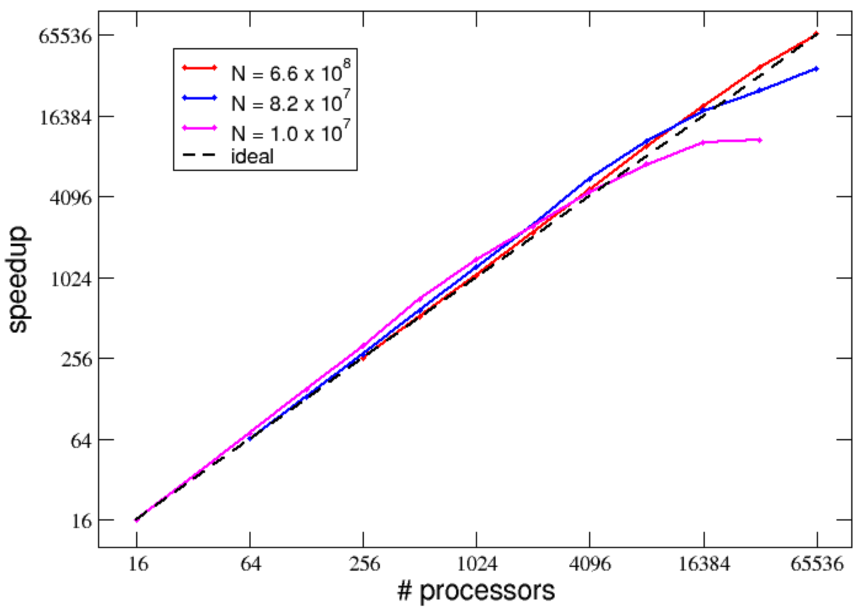
- Time structure

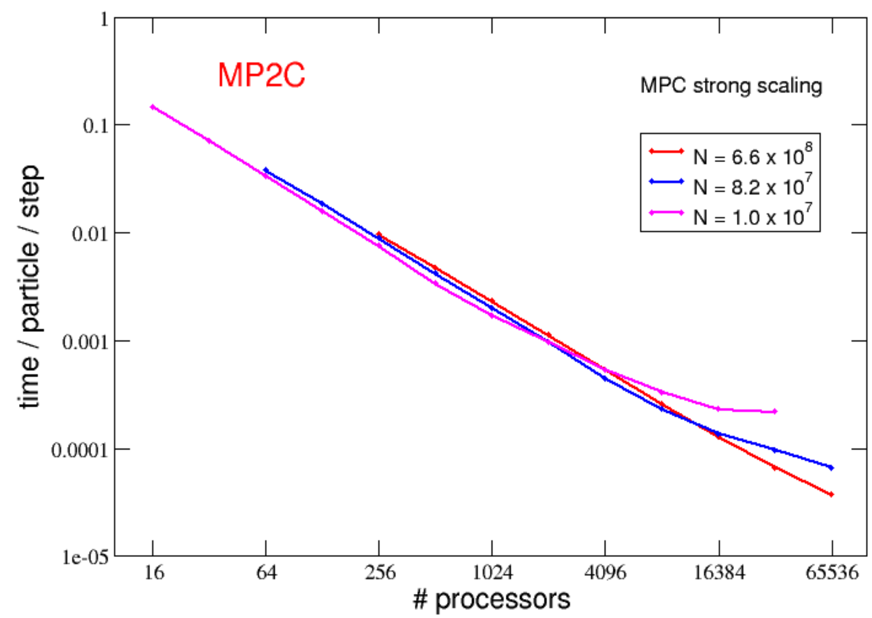




MPC: Strong scaling

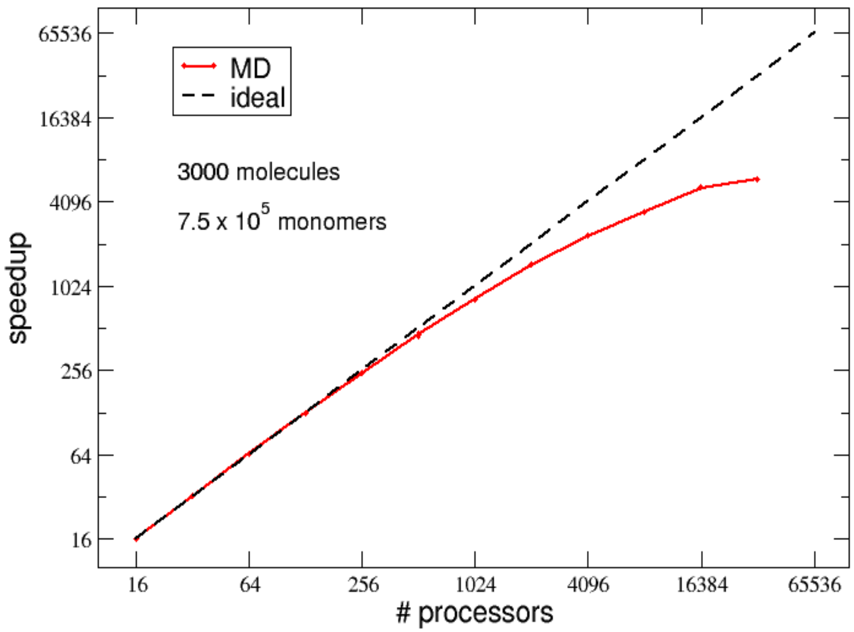
- Keep total number of particles constant

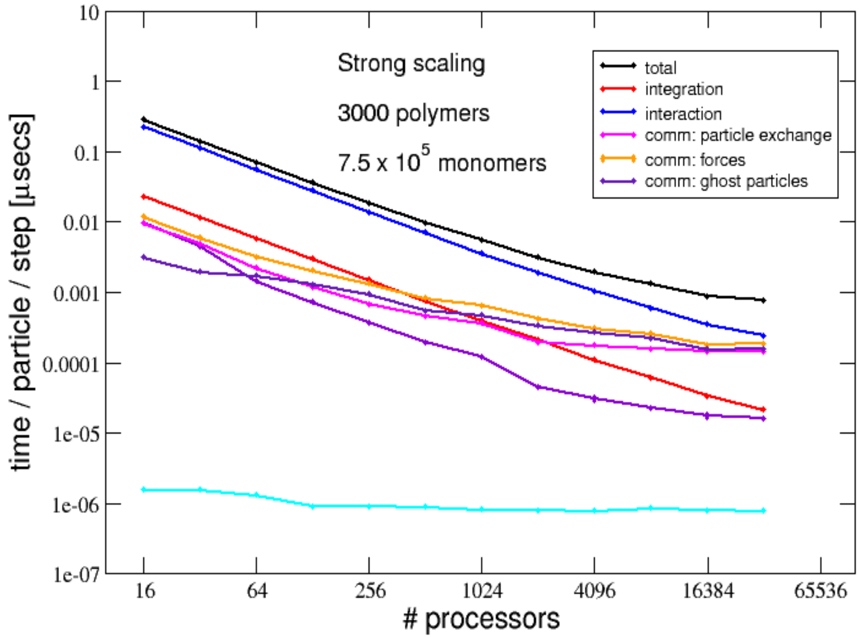




MD: Strong scaling

- Keep total number of particles constant
- Benchmark system: 3000 Polymer chains with 250 monomers





Summary

- Good scaling properties of hydrodynamic part of MP2C due to homogenous distribution and local collisions of particles
- Although not bad, MD part needs some improvement for inhomogenous distributions and low density systems
- To be done:
  - Load-balancing
  - Improvement of I/O
    - at the moment well below bandwidth capacity
    - solution with PNetCDF or SION